AD-A153 158   MAXIMUM FLOW IN PLANAR NETWORKS WITH EXPONENTIALLY   1/1
               DISTRIBUTED ARC CAPACI..(U) NORTH CAROLINA UNIV AT
               CHAPEL HILL CURRICULUM IN OPERATIONS R..
UNCLASSIFIED   V G KULKARNI ET AL. DEC 84   F/G 12/2   NL
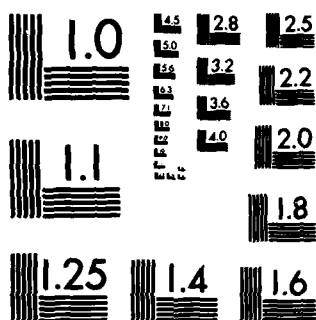
END
FILMED

DTIC

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(4)

OPERATIONS RESEARCH AND SYSTEMS ANALYSIS

Maximum Flow in Planar Networks with
Exponentially Distributed Arc Capacities

by

V.G. Kulkarni
Curriculum in Operations Research and
Systems Analysis

V.G. Adlakha
Department of Management
University of Baltimore
Baltimore, MD 21201

UNIVERSITY OF NORTH CAROLINA
AT CHAPEL HILL

85 4 2 110

Maximum Flow in Planar Networks with
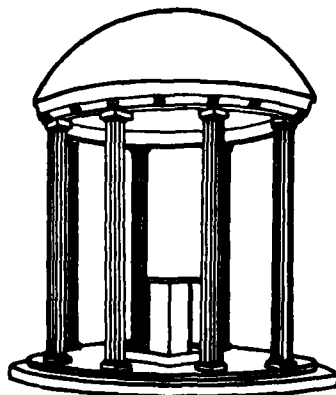Exponentially Distributed Arc Capacities

by

V.G. Kulkarni
Curriculum in Operations Research and
Systems Analysis

V.G. Adlakha
Department of Management
University of Baltimore
Baltimore, MD 21201

Technical Report No. UNC/ORSA/TR-84/15

December 1984

DTIC
ELECTE
S APR 26 1985
D
_ B

AIR FORCE OFFICE ⸳⸳⸳⸳⸳⸳⸳⸳⸳⸳⸳⸳⸳⸳⸳⸳ (AFSC)
NOTICE ⸳⸳ ⸳⸳⸳
Th⸳⸳   ⸳⸳⸳
a⸳⸳.
Dist⸳⸳⸳
MATTHE⸳ ⸳
Chief, Techni⸳⸳⸳ ⸳⸳⸳ ⸳⸳⸳⸳⸳⸳⸳⸳ Division

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
UNCLASSIFIED	

2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
UNC/ORSA/TR-84/15	AFOSR-TR- 85-0286

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
University of North Carolina		Air Force Office of Scientific Research

6c. ADDRESS (City, State and ZIP Code)	7b. ADDRESS (City, State and ZIP Code)
Curriculum in Operations Research and Systems Analysis, Smith Bldg 128A, Chapel Hill NC 27514	Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332-6448

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
AFOSR	NM	AFOSR-84-0140

8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
Bolling AFB DC 20332-6448	61102F	2304	A5	

11. TITLE (Include Security Classification)
MAXIMUM FLOW IN PLANAR NETWORKS WITH EXPONENTIALLY DISTRIBUTED ARC CAPACITIES

12. PERSONAL AUTHOR(S)
V.G. Kulkarni and V.G. Adlakha

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Technical	FROM _____ TO _____	DEC 84	37

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Maximum flow; stochastic networks; multi-state reliability modeling; Markov chains.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper develops methods for the exact computation of the distribution of the maximum flow and related quantities in a planar network with independent and exponentially distributed arc capacities. A continuous time Markov chain (CTMC) with upper triangular rate matrix and single absorbing state is constructed with the property that the time until absorption in this absorbing state is equal to the value of the maximum flow in the network. Recursive algorithms are developed for computing the distribution and moments of the maximum flow. Algorithms are also presented to compute the probability that a given cut is the minimum capacity cut in the network. The algorithms are efficient and computationally stable. Distribution of the maximum flow, given a minimum cut, is studied.

----- cont. keywords include : -

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
MAJ Brian W. Woodruff	(202) 767- 5027	NM

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE.

85 4 2 110

Abstract

This paper develops methods for the exact computation of the distribution of the maximum flow and related quantities in a planar network with independent and exponentially distributed arc capacities. A continuous time Markov chain (CTMC) with upper triangular rate matrix and single absorbing state is constructed with the property that the _time_ until absorption in this absorbing state is equal to the _value_ of the maximum flow in the network. Recursive algorithms are developed for computing the distribution and moments of the maximum flow. Algorithms are also presented to compute the probability that a given cut is the minimum capacity cut in the network. The algorithms are efficient and computationally stable. Distribution of the maximum flow, given a minimum cut, is studied.

1. Introduction

In this paper we investigate directed planar stochastic networks and consider the problem of computing the probability distribution of the maximum flow that can be sent from the source s to the sink t without violating the capacity constraints of the arcs. Among several situations where the maximum flow problem in stochastic networks is encountered in practice, we describe two.

First, consider a network with deterministic arc capacities. Assume that the existing flow in the network is random, thus the excess capacity available on each arc is random. We are interested in determining how much additional flow can be sent from the source to the sink in this network. Thus we are faced with the maximum flow problem in a stochastic network where the arc capacities are the random excess capacities in the original network. Such probabilistic capacitated networks may arise in communication or transportation applications (see Frank and Frisch [1971]).

Second, consider a system of components which can be represented by a network. Each arc of the network represents a component in the system. Each component can be either "up" or "down". The system as a whole is "up" if and only if there is a path of "up" components from the source to the sink in the network representation of the system. If we define the capacity of an arc to be 0 or 1 depending upon whether the component represented by the arc is "down" or "up" we get a stochastic network. The probability that the system is "up" is the probability that the maximum flow in the stochastic network is nonzero. In fact the actual value of the maximum flow can be taken to indicate the "health" of the system: the higher the maximum flow, the "healthier" the system.

When we consider multi-state multi-component systems, i.e. systems where the components as well as the system may exist in more than two states of degradation, the maximum flow concept provides a natural extension of the binary case. We construct a stochastic network where the capacity of an arc is a monotonic function of the state of the component it represents. The state of the system is then defined as the maximum flow from the source to the sink in this network.

The value of the maximum flow in a stochastic network is a random variable. We are interested in its distribution, moments and its dependence on the network characteristics. Frank and Frisch [1971] have offered a formulation utilizing transforms. The procedures, though conceptually simple, are impractical from the computational point of view, even in the special case of normally distributed arc capacities. Evans [1976] has addressed the problem for a network with integer-valued, discretely distributed arc capacities.

Comprehensive procedures have been developed for the maximum flow problem when arc capacities are deterministic. Ford and Fulkerson [1962] developed a procedure and showed that the value of the maximum flow from the source s to the sink t is equal to the capacity of the minimum capacity (s,t) cut in the network. When the network is (s,t) planar, i.e., it has a planar graph representation such that an arc from t to s can be drawn without violating planarity, one can draw its dual network. The shortest path problem for this dual network can then be solved by using one of the standard shortest path algorithms (see Dijkstra [1959], Elmaghraby [1977]). The duality theory asserts that the

length of the shortest path in the dual network is equal to the
capacity of the minimum capacity cut in the original network, and
thus provides the value of the maximum flow. Alternate
algorithms for the maximum flow in (s,t) planar networks, which
avoid constructing the dual, are described in Itai and Shiloach
[1979].

In this paper, we consider the maximum flow problem in (s,t)
planar networks where arc capacities are independent
exponentially distributed random variables. When the capacities
are not exponentially distributed one can approximate their
actual distributions by phase type distributions. The methods
developed in this paper can be directly extended to the case when
the arc capacities have phase type distributions. We do not
discuss this extension in this paper since the most significant
contribution of this paper is the construction of the stochastic
processes which facilitate the solution of the maximum flow
problem.

In Section 2 we establish the network terminology related
to (s,t) planar networks. In particular we introduce the
"topmost path first" order on the set of all (s,t) paths in the
network. In Section 3 we present an algorithm for the maximum
flow problem in (s,t) planar networks with deterministic arc
capacities, which is subsequently used in the development of our
algorithms.

In Section 4 we construct a finite state space continuous
time Markov chain (CTMC) with the following property: the time
until absorption in the absorbing state in this Markov chain is
equal to the value of the maximum flow in the network. The state

space is shown to be the set of (s,t) paths in the network augmented by an absorbing state. When these paths are ordered according to the "topmost path first" order, the rate matrix of the CTMC becomes upper triangular. This structure yields computationally simple algorithms to compute the distribution and moments of the maximum flow. These algorithms are described in Section 5.

In Section 6 we introduce the criticality indicator of a minimal (s,t) cut, which takes the value 1 if the cut is a minimum capacity cut in the network, and 0 otherwise. The criticality index of a cut is defined to be the probability that the criticality indicator of the cut is 1. We study the joint distribution of the criticality indicator of a cut and the maximum flow and develop algorithms to compute the criticality index of a cut.

In Section 7 we document the computational results using four networks as examples. The conclusions are presented in Section 8.

2. Planar Networks

Let $G = (V,A)$ be a network with vertex set V, arc set A, source node s, sink node t, and N paths. It is assumed that G is (s,t) planar and that one planar representation of G is fixed in advance.

An arc $e \in A$ is <u>incident</u> on a node u if u is the starting or ending node of e. Let $E(u)$ denote the set of arcs incident on u. Each arc $e \in E(u)$ induces a partial order, on $E(u)$ as follows. Arc e_1 is said to be <u>before</u> e_2 <u>in the partial order induced by</u> e <u>on</u> $E(u)$ if e_1, $e_2 \in E(u)$ and e_1 appears before e_2 in a clockwise sweep around u starting from e.

A path in G is described by the sequence of nodes it visits. A path is called simple if it does not visit any node more than once. Let $P_1 = (v_0 = s, v_1, \ldots, v_m = t)$ and $P_2 = (u_0 = s, u_1, \ldots, u_n = t)$ be two simple (s,t) paths in G. Assume that an imaginary arc (t,s) is drawn on the planar representation of G without violating the planarity. Let $v_{+1} = t$. The path P_1 <u>lies above</u> P_2 (or, equivalently, P_2 <u>lies below</u> P_1) if there exists an index r such that $v_i = u_i$ for $i = 1, 2, \ldots, r$ and $v_{r+1} \neq u_{r+1}$ and the arc (v_r, v_{r+1}) is before (u_r, u_{r+1}) in the partial order induced by (v_{r+1}, v_r) on $E(v_r)$.

The relation "<u>lies above</u>" defines a partial order on the set of all (s,t) paths in G. Let L be the ordered set of paths $\{P_1, P_2, \ldots, P_N\}$ such that if P_i lies above P_j then $i < j$. The paths in L are said to be in the topmost path first (TPF) order. An algorithm to list the paths in the TPF order is described

below. It is a modified version of the backtracking algorithm described in Reed and Tarjan [1975].

Algorithm to List Paths in TPF Order:

Step 0. Set $v_{-1} = t$.

All nodes unlabelled.

All arcs unscanned.

Step 1. Set $i=0$, $v_0 = s$. Label s.

Step 2. Let e be the first unscanned arc leaving v_i that appears in the clockwise sweep around v_i starting from (v_{i-1}, v_i) such that head node of e is unlabelled.

If no such e exists, go to Step 4.

Step 3: Set v_{i+1} = head node of e, label v_{i+1}. Scan e.

If $v_\cdot = t$, go to Step 5.

else, set $i=i+1$ and go to Step 2.

Step 4. If $i=0$, stop. (All paths are listed).

Else, unscan all arcs leaving v_i, unlabel v_i.

Set $i=i-1$ and go to Step 2.

Step 5. An (s,t) path of length $i+1$ is found.

List the nodes $v_0, v_1, \ldots, v_i$.

Unlabel t, set $i=i-1$ and go to Step 2.   □

Choice of e in Step 2 guarantees that the paths are generated in the TPF order.

Let $P_1 = (v_0 = s, v_1, \ldots, v_m = t)$ and $P_2 = (u_0 = s, u_1, \ldots, u_n = t)$ be two simple (s,t) paths in G. The path P_1 lies completely above P_2 (or, equivalently, P_2 lies completely below P_1) if $v_i = u_j$ and v_{i+1}

$\nmid u_{j+1}$ implies that (v_i, v_{i+1}) is before (u_j, u_{j+1}) in the partial order induced by (v_{i-1}, v_i) on $E(v_i)$. For each P in L and an arc e on P, define an alternate path P(e) to be the first path in L that does not use arc e and lies completely below P. If no such path exists we say $P(e) = \phi$. An algorithm to construct P(e) given P and e is described in Itai and Shiloach [1979].

We end this section with an example which illustrates the concepts defined here. Consider the planar network of six nodes and nine arcs displayed in Figure 1. Node 1 is the source and node 5 is the sink. Table I lists the paths in this network in TPF order. Path (1,3,4,5) lies below path (1,2,3,6,5) but does not lie completely below. Path (1,3,4,5) lies completely below path (1,2,3,4,5). The alternate path for path (1,2,4,6,5) is (1,2,3,4,6,5) if arc e is (2,4). There is no alternate path for path (1,2,4,6,5) if arc e is (6,5).

3. Maximum Flow in Deterministic Planar Networks

In this section we describe an algorithm to find the maximum flow from s to t in a planar network. The algorithm is suggested in Ford and Fulkerson [1956], developed in Berge and Ghouila-Houri [1962] and its time complexity is reduced to $O(|V| \log |V|)$ by Itai and Shiloach [1979]. The algorithm is described below since it is the starting point for our analysis in the stochastic case.

Let $c(e)$ be the non negative deterministic flow capacity of arc e. Let f^* be the maximum flow that can go from s to t. As before let L be the list of (s,t) paths in the TPF order. Define the capacity of a path $P \epsilon L$ as

$$c(P) = \min_{e \text{ on } P} \{c(e)\}. \qquad (3.1)$$

Maximum Flow Algorithm:

Step 1. Let P be the first path in L. Set $f^* = 0$.

Step 2. $f^* = f^* + c(P)$

$c(e) = c(e) - c(P)$ for all $e \epsilon P$.

Step 3. Let P' be the next path in L such that $c(P') > 0$.

If no such path exists, go to Step 4.

Else, set $P \leftarrow P'$ and go to Step 2.

Step 4. Stop, f^* is the maximum flow.   □

Remark 1. This algorithm is called the "path filling" algorithm, since it starts with the first path in L and keeps filling them to capacity until no more path can be found. The proof that this algorithm produces the maximum flow is given by both Berge and Ghouila Houri [1962] and Itai and Shiloach [1979].

Remark 2. In Step 2, let e be that arc on P for which c(P) = c(e). If this e is unique, then it can be shown that P' is the alternate path P(e) as defined in Section 2.

Remark 3. The path filling algorithm works if the paths are used in TPF (or the reverse of TPF) order. An arbitrary order may not work, i.e. all the paths may get saturated while the flow is not maximum.

Remark 4. The TPF order is not defined for a non planar network In such a case it is possible to construct an example where the path filling algorithm fails no matter what ordering of paths is used.

4. Maximum Flow in Stochastic Planar Networks

Let G = (V,A) be the (s,t) planar network with source s and sink t and L be the ordered set of (s,t) paths in G in TPF order. Let C(e) be a non-negative random variable representing the capacity of arc e. Let T be the maximum flow that can be sent from s to t. We are interested in the distribution and moments of T. To analyze this problem we construct a stochastic process with a single absorbing state so that the time until absorption in this state is equal to the maximum flow, T, from s to t.

Visualize the network G as a network of pipelines, with arc e capable of handling C(e) gallons of flow per unit time. Suppose that there is a pump at node s that forces fluid into the network at the rate of t gallons per unit time at time t. Thus the flow of the pump increases linearly with time. The nodes act as multiple-valves controlled by a complex mechanism which operates as follows: At time t = 0, the first path in L, call it P, is open for the fluid flow; all other paths are blocked. When this path fills up due to capacity restriction of arc e on P, the flow is then diverted to the alternate path P(e). This process continues as long as there is an unfilled alternate path available. When all paths are filled the pump flow cannot be increased any further.

Define X(t) to be the path in which flow is increasing at time t. If at time t the network is saturated, i.e. the flow is not increasing, X(t) is defined to be ϕ. X(t) represents the state of the system at time t. Note that the network handles a flow of t gallons/unit time at time t. From the construction of

X(t) and the path filling algorithm of Section 3, it is clear that

$$T = \min_{t \geq 0} \{X(t) = \phi\}. \qquad (4.1)$$

From now on we assume the following:

A1. $\{C(e), e\epsilon A\}$ is a set of independent random variables.

A2. $C(e)$ is exponentially distributed with parameter

$\mu(e) > 0$ (mean $= 1/\mu(e)$).

Theorem 1. Under assumptions A1 and A2, $\{X(t), t \geq 0\}$ is a CTMC with state space $L^* = L \cup \{\phi\}$ and rate matrix $Q = [q(P,P')]$, $P, P' \epsilon L^*$, where

$$q(P,P') = \sum_{e \text{ on } P: P(e) = P'} \mu(e) \qquad \text{if } P' \neq P; \; P,P' \epsilon L$$

$$= -\sum_{e \text{ on } P} \mu(e) \qquad \text{if } P = P' \epsilon L \qquad (4.2)$$

$$= 0 \qquad \text{if } P = \phi.$$

Proof: The fact that $\{X(t), t \geq 0\}$ is a CTMC follows from assumptions A1 and A2. Now suppose that $X(t) = P \epsilon L$. If the arc e on P saturates, which happens at rate $\mu(e)$, the state changes to $P(e)$. If $P(e)$ does not exist, the state changes to ϕ. If $X(t) = \phi$ it remains ϕ from then on. The equations for $q(P,P')$ follow from these observations.   □

Remark 5. From the construction of $\{X(t), t \geq 0\}$ it is obvious that the process does not visit the same state twice, since once a path is saturated it stays saturated. Hence the rate matrix is upper triangular. This structural property is critical in

developing simple algorithms for computing the distribution and moments of T.

Example. Consider the example network of Figure 1. The state space for the CTMC representation for this Markov chain is partly given in Table I. (Table I gives L, the state space is $L^* = L \cup \{\phi\}$). The rate matrix for this problem is given in Table II.

5. Distribution and Moments of Maximum Flow

In this section we develop algorithms to compute the
distribution and moments of the maximum flow. It is possible to
develop two types of algorithms, called backward and forward
algorithms. We present both versions, but we illustrate only the
backward algorithms by means of an example. Also, from now on we
identify the set L with the set $\{1,2,\ldots,N\}$, i.e. $X(t) = i$ means
that the state of the system at time t is given by the i-th path
in L. The state ϕ will be identified by the number N+1. Thus
equation (4.1) becomes

$$T = \min \{t \geq 0: X(t) = N+1 \mid X(0) = 1\}. \qquad (5.1)$$

We are interested in the distribution of maximum flow

$$F(t) = P(T \leq t) \qquad (5.2)$$

and the k-th moment

$$m(k) = E(T^k). \qquad (5.3)$$

5.1 Distribution of Maximum Flow

Backward Algorithm: Define

$$p_i(t) = P\{X(t) = N+1 \mid X(0) = i\}, \quad 1 \leq i \leq N + 1. \qquad (5.4)$$

Then $F(t) = p_1(t)$. The differential equations for $p_i(t)$ are
given by

$$p_i'(t) = \sum_{j \geq i} q_{ij} p_j(t)$$

$$1 \leq i \leq N+1 \qquad (5.5)$$

$$p_i(0) = \delta_{i,N+1}$$

where $\delta_{i,j} = 1$ if $i = j$, 0 otherwise. Due to the upper triangular nature of Q, equations (5.5) can be solved in a backward manner, starting with $p_{N+1}(t) = 1$ for $t \geq 0$, and computing $p_N(t),\ldots,p_1(t) = F(t)$ in that order.

Example. For network 1, equations (5.5) become

$$p_9'(t) = 0,$$

$$p_8'(t) = -(\mu_2+\mu_6+\mu_9)p_8(t) + (\mu_2+\mu_6+\mu_9)p_9(t),$$

$$p_7'(t) = -(\mu_2+\mu_5+\mu_8+\mu_9)p_7(t) + (\mu_5+\mu_8)p_8(t) + (\mu_2+\mu_9)p_9(t),$$

$$\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad (5.6)$$

$$p_3'(t) = -(\mu_1+\mu_4+\mu_5+\mu_7)p_3(t) + \mu_7 p_4(t) + \mu_5 p_5(t) + (\mu_1+\mu_4)p_6(t),$$

$$p_2'(t) = -(\mu_1+\mu_3+\mu_8+\mu_9)p_2(t) + (\mu_3+\mu_8)p_5(t) + \mu_1 p_7(t) + \mu_9 p_9(t),$$

$$p_1'(t) = -(\mu_1+\mu_3+\mu_7)p_1(t) + \mu_7 p_2(t) + \mu_3 p_3(t) + \mu_1 p_6(t).$$

Forward Algorithm: Define

$$\bar{p}_j(t) = P\{X(t) = j \mid X(0) = 1\}, \qquad 1 \leq j \leq N + 1. \quad (5.7)$$

Then $F(t) = \bar{p}_{N+1}(t)$. The differential equations for $\bar{p}_j(t)$ are given by

$$\bar{p}_j(t) = \sum_{i \leq j} \bar{p}_i(t)q_{ij}$$

$$1 \leq j \leq N + 1 \qquad\qquad (5.8)$$

$$\bar{p}_j(0) = \delta_{j,1}.$$

Again since Q is upper triangular, the above equations can be solved in a forward manner, starting with $\bar{p}_1(t) = \exp(q_{11}t)$, $t \geq 0$, and computing $\bar{p}_2(t),\ldots,\bar{p}_{N+1}(t) = F(t)$ in that order.

Numerical Evaluation of F(t): Uniformization technique described in Ross [1983] is used to numerically evaluate $F(t)$. The details

of this technique are given in Kulkarni and Adlakha [1984].   Here
we give only the final result.   Define

$$q = \max_{1 \leq i \leq N+1} \{-q_{ii}\} \tag{5.9}$$

and let

$$q_{ij}^* = \delta_{ij} + q_{ij}/q, \qquad 1 \leq i, j \leq N+1 \tag{5.10}$$

Let $Q^* = [q_{ij}^*]$ and let α_n be the (1,N+1)th element of the nth
power of Q^*.   It is easy to see that α_n's increase to 1 as $n \to \infty$.
Then we have

$$F(t) = \sum_{n=0}^{\infty} \alpha_n e^{-qt}(qt)^n/n! . \tag{5.11}$$

In numerical calculations we choose an $\epsilon > 0$ and an M such that
$\alpha_M \geq 1-\epsilon$.   Then compute

$$F_M(t) = 1 - \sum_{n=0}^{M} (1-\alpha_n)e^{-qt}(qt)^n/n! \tag{5.12}$$

and

$$\tilde{F}_M(t) = \alpha_M - \sum_{n=0}^{M} (\alpha_M-\alpha_n)e^{-qt}(qt)^n/n! . \tag{5.13}$$

It can be shown easily that $F_M(t)$ and $\tilde{F}_M(t)$ provide lower and upper
bounds for F(t) within ϵ.   It is a straightforward matter to
compute α_n's sequentially.   Thus uniformization is a
computationally stable and rapid method of computing F(t).

5.2 Moments of Maximum Flow

We now describe the backward and forward algorithms to compute m(k), the k-th moment of T.

Backward Algorithm: Define

$$T_i = \min \{t \geq 0: X(t) = N+1 \mid X(0) = i\}, \quad 1 \leq i \leq N + 1 \quad (5.14)$$

and

$$m_i(k) = E(T_i^k), \qquad\qquad 1 \leq i \leq N + 1. \quad (5.15)$$

As $\{X(t), t \geq 0\}$ gets absorbed in the state $N + 1$ with probability 1, $m_i(0) = 1$ for $1 \leq i \leq N + 1$. Also, for $k \geq 1$, $m_{N+1}(k) = 0$. It can be shown that

$$m_i(k) = [k m_i(k-1) + \sum_{j>i} q_{ij} m_j(k)]/q_i \quad (5.16)$$

where $q_i = - q_{ii}$. We have $m(k) = m_1(k)$. To compute m(k) one needs to compute $m_i(r)$ for $r = 1, 2, \ldots, k$; $i = N + 1, N, \ldots, 2, 1$ in that order.

Example. For network 1 equations (5.16) become

$$m_9(k) = 0,$$

$$m_8(k) = [k m_8(k-1) + (\mu_2 + \mu_6 + \mu_9) m_9(k)]/(\mu_2 + \mu_6 + \mu_9),$$

$$m_7(k) = [k m_7(k-1) + (\mu_5 + \mu_8) m_8(k) + (\mu_2 + \mu_9) m_9(k)]/(\mu_2 + \mu_5 + \mu_8 + \mu_9),$$

$$\vdots \qquad\qquad (5.17)$$

$$m_3(k) = [k m_3(k-1) + \mu_7 m_4(k) + \mu_5 m_5(k) + (\mu_1 + \mu_4) m_6(k)]/(\mu_1 + \mu_4 + \mu_5 + \mu_7),$$

$$m_2(k) = [k m_2(k-1) + (\mu_3 + \mu_8) m_5(k) + \mu_1 m_7(k) + \mu_9 m_9(k)]/(\mu_1 + \mu_3 + \mu_8 + \mu_9),$$

$$m_1(k) = [k m_1(k-1) + \mu_7 m_2(k) + \mu_3 m_3(k) + \mu_1 m_6(k)]/(\mu_1 + \mu_3 + \mu_7).$$

Forward Algorithm: Define

$$\bar{T}_j = \inf\{t \geq 0 : X(t) = j \mid X(0) = 1\} \tag{5.18}$$

where the infimum over an empty set is defined to be $+ \infty$. Let

$$\bar{m}_j(k) = E(\bar{T}_j^k), \qquad k \geq 0. \tag{5.19}$$

It is possible to have $\bar{m}_j(0) = P(\bar{T}_j < \infty) < 1$ for $1 \leq j \leq N$. It can be shown that

$$\bar{m}_j(k) = \sum_{i<j} [\sum_{r=0}^{k} (k!/r!)(\bar{m}_i(r)/q_i^{k-r})](q_{ij}/q_i)$$

$$\bar{m}_1(k) = \delta_{k,0} \qquad\qquad k \geq 0, \ 1 \leq j \leq N+1. \tag{5.20}$$

These equations are solved recursively in a forward manner. It is clear that equations (5.16) can be solved more efficiently than equations (5.20).

6. Joint Performance and Criticality Index

Let $Y \subset A$ be a prespecified (s,t) cut. Assume that Y is a minimal cut, i.e., no proper subset of Y is a (s,t) cut. Then it is known that there is a set $B \subset V$ such that

$$Y = C(B,\bar{B}) = \{(u,v) \in A: u \in B, v \in \bar{B}\}. \tag{6.1}$$

Define the capacity of a cut to be the sum of the capacities of its constituent arcs. Let

$$I(Y) = 1 \quad \text{if } Y \text{ is minimum capacity } (s,t) \text{ cut in } G$$

$$= 0 \quad \text{otherwise.} \tag{6.2}$$

$I(Y)$ is called the criticality indicator of the cut Y. Define the criticality index of the cut Y as

$$R(Y) = P(I(Y) = 1). \tag{6.3}$$

In this section we study the joint distribution of $I(Y)$ and the maximum flow as well as the criticality index $R(Y)$. The key to the development of these algorithms is the following characterization of minimum capacity cut from Ford and Fulkerson [1962]:

A cut $Y = C(B,\bar{B})$ is a minimum capacity (s,t) cut if and only if the arcs in Y are saturated while the arcs in $C(\bar{B},B)$ are empty when the flow from s to t is maximum.

A (s,t) path in G is called Y-admissible if it intersects Y in one and only one arc. It is clear that a path is Y-admissible if and only if it has no arcs from $C(\bar{B},B)$. It can be easily seen that the first and the last paths in L are Y-admissible for any minimal cut Y. Let $L(Y)$ be the ordered set of Y-admissible paths obtained by deleting all non Y-admissible paths from L.

We now modify the pipe filling process described in Section 4 as follows: Consider the following cases when the current path, say P, fills up due to the capacity restriction of arc e on P:

(1) $e \varepsilon Y$, $P(e) = \phi$,

(2) $e \not\in Y$, $P(e) = \phi$,

(3) $e \varepsilon Y$, $P(e) \neq \phi$, $P(e)$ is Y-admissible,

(4) $e \not\in Y$, $P(e) \neq \phi$, $P(e)$ is Y-admissible, $Y \cap P = Y \cap P(e)$,

(5) $e \not\in Y$, $P(e) \neq \phi$, $P(e)$ is Y-admissible, $Y \cap P \neq Y \cap P(e)$,

(6) $e \not\in Y$, $P(e) \neq \phi$, $P(e)$ is not Y-admissible.

It can be shown that the case "$e \varepsilon Y$, $P(e) \neq \phi$, $P(e)$ is not Y-admissible" cannot occur if Y is a minimal cut. The original pipe filling process described in Section 4 terminates if cases 1 or 2 are encountered. The modified process terminates if cases 1, 2, 5 or 6 are encountered. If the modified process terminates due to case 1, it will be called a "good" termination; else it will be called a "bad" termination. In cases 3 or 4 the process continues by diverting the flow to the path $P(e)$.

Let $X_Y(t)$ represent the path in which flow is increasing at time t in the modified process. If the flow is not increasing in any path at time t, the process must have terminated before t. If the process ends with a "good" ("bad") termination at time τ, $X_Y(t)$ is defined to be $\phi (\phi')$ for $t \geq \tau$. It is clear that as long as $X_Y(t) \varepsilon L$, it is in fact in $L(Y)$, since $X_Y(0)$ is a Y-admissible path. Thus $\{X_Y(t), t \geq 0\}$ is a stochastic process with state space

$$L^*(Y) = L(Y) \cup \{\phi, \phi'\}. \tag{6.4}$$

The following theorem brings out the significance of the process $\{X_Y(t), t \geq 0\}$.

<u>Theorem 2</u>. Let $\{X_Y(t), t \geq 0\}$ be the modified pipe filling process on $L^*(Y)$ with $X_Y(0)$ = first path in $L(Y)$. Let

$$T(Y) = \min \{t \geq 0 : X_Y(t) = \phi\}. \qquad (6.5)$$

Then

$$P(T(Y) \leq t) = P(I(Y) = 1 \text{ and capacity of } Y \leq t). \qquad (6.6)$$

<u>Proof</u>: We shall show that

$$\{X_Y(t) = \phi\} = \{I(Y) = 1 \text{ and capacity of } Y \leq t\}. \qquad (6.7)$$

($\Rightarrow$) Suppose $X_Y(t) = \phi$. Then, from the construction of the modified pipe filling process, maximum flow of value $\leq t$ has been achieved; no arcs of $C(\bar{B}, B)$ have been used and the arcs in Y are full. Then from the characterization of max-flow by min-cut, the cut Y is the minimum capacity cut and its capacity (which is same as the maximum flow) is $\leq t$. Thus $I(Y) = 1$ and capacity of $Y \leq t$.

($\Leftarrow$) Suppose $I(Y) = 1$ and capacity of $Y \leq t$. Then from the characterization of max-flow by min-cut the arcs in $C(\bar{B}, B)$ are empty, the arcs in Y are saturated and the maximum flow $\leq t$. Hence, $X_Y(t)$ must have stayed in $L(Y)$ until it hit ϕ before t. Thus $X_Y(t) = \phi$.

As ϕ is an absorbing state

$$P(T(Y) \leq t) = P(X_Y(t) = \phi). \qquad (6.8)$$

The result now follows from equations (6.7) and (6.8). □

As ϕ' is also an absorbing state, $T(Y)$ is an incomplete random variable. In fact

$$R(Y) = P(I(Y) = 1) = P(T(Y) < \infty). \qquad (6.9)$$

Thus, the criticality index of a cut Y is given by the probability that the process $\{X_Y(t), t \geq 0\}$ gets absorbed in ϕ.

- 21 -
/segNote that Theorem 2 is true regardless of the assumptions A1 and A2. If A1 and A2 hold, $\{X_Y(t), t \geq 0\}$ can be shown to be a CTMC as proved in the following theorem. First, we introduce some notation.

If $X_Y(t) = P \epsilon L(Y)$ and arc e on P saturates, a transition takes place to some other path $P' \epsilon L(Y)$, ϕ or ϕ'. This transition is denoted by (P,e) and belongs to one of the six cases described earlier. For $P \epsilon L(Y)$, define

$$M(P,P') = \{e \text{ on } P: P' = P(e), \text{ the transition}$$
$$(P,e) \text{ belongs to case 3 or 4}\}, \qquad (6.10)$$

$$M(P,\phi) = \{e \text{ on } P: \text{ the transition } (P,e)$$
$$\text{belongs to case 1}\}, \qquad (6.11)$$

$$M(P,\phi') = \{e \text{ on } P: \text{ the transition } (P,e)$$
$$\text{belongs to case 2, 5 or 6}\}. \qquad (6.12)$$

Theorem 3. Under assumptions A1 and A2, $\{X_Y(t), t \geq 0\}$ is a CTMC on $L^*(Y)$ with the rate matrix $Q(Y) = [q_Y(P,P')]$ $(P,P' \epsilon L^*(Y))$, defined as follows

$$
\begin{aligned}
q_Y(P,P') &= \sum_{e \epsilon M(P,P')} \mu(e) && \text{if } P,P' \epsilon L(Y), P \neq P' \\
&= \sum_{e \epsilon M(P,\phi)} \mu(e) && \text{if } P' = \phi, P \epsilon L(Y) \\
&= \sum_{e \epsilon M(P,\phi')} \mu(e) && \text{if } P' = \phi', P \epsilon L(Y) \quad (6.13) \\
&= -\sum_{e \text{ on } P} \mu(e) && \text{if } P' = P \epsilon L(Y) \\
&= 0 && \text{if } P = \phi \text{ or } \phi'.
\end{aligned}
$$

Proof: Follows from the construction of $\{X_Y(t), t \geq 0\}$ and the definitions of $M(P,P')$, $M(P,\phi)$ and $M(P,\phi')$. □

Let the paths in $L(Y)$ be numbered $1,2,\ldots,m$ and the states ϕ and ϕ' be identified by the numbers $m+1$ and $m+2$ respectively. Then from Theorem 2 we get

$$P(I(Y) = 1 \text{ and capacity of } Y \leq t)$$

$$= P(I(Y) = 1 \text{ and maximum flow} \leq t)$$

$$= P(X_Y(t) = m+1 \mid X_Y(0) = 1). \tag{6.14}$$

Thus the joint distribution of the maximum flow and the criticality indicator of the cut Y can be obtained as the distribution of time until absorption in the state $m+1$ of the modified CTMC $\{X_Y(t), t \geq 0\}$ starting from state 1. The uniformization technique of Section 5 can be used to compute this distribution numerically.

The criticality index of the cut Y is given by

$$R(Y) = P(X_Y(t) = m+1 \text{ for some } t \geq 0 \mid X_Y(0) = 1). \tag{6.15}$$

Thus the criticality index can be computed as the probability that the process $\{X_Y(t), t \geq 0\}$, starting in state 1, gets absorbed in state ϕ. The backward and forward algorithms for doing this are described below. We write $q_{ij}(Y)$ for $q_Y(P,P')$ where P and P' are identified by numbers i and j respectively.

Backward Algorithm: Define

$$v_i = P\{X_Y(t) = m+1 \text{ for some } t \geq 0 \mid X_Y(0) = i\}, \quad 1 \leq i \leq m+1. \tag{6.16}$$

Then

$$v_i = \sum_{j > i} v_j q_{ij}(Y)/q_i(Y) \qquad 1 \leq i \leq m \tag{6.17}$$

where $q_i(Y) = -q_{ii}(Y)$ and $v_{m+1} = 1$, $v_{m+2} = 0$. These quantities can be computed in a backward manner to yield $R(Y) = v_1$.

Example. Consider network 1 and let Y be the minimal cut $\{3,5,9\}$. For this case, $B = \{1,2,3,6\}$ and $C(\bar{B},B) = \{8\}$. The

set of Y-admissible paths, L(Y), is shown in Table III. The Q(Y) matrix is shown in Table IV. Equations (6.17) for this problem are as follows:

$$v_7 = 0,$$
$$v_6 = 1,$$
$$v_5 = \mu_9 v_6/(\mu_2+\mu_6+\mu_9),$$
$$v_4 = \mu_5 v_5/(\mu_2+\mu_5+\mu_7),$$
$$v_3 = ((\mu_1+\mu_4)v_5+\mu_9 v_6)/(\mu_1+\mu_4+\mu_6+\mu_9),$$
$$v_2 = (\mu_5 v_3+(\mu_1+\mu_4)v_4)/(\mu_1+\mu_4+\mu_5+\mu_7),$$
$$v_1 = \mu_2 v_2/(\mu_1+\mu_3+\mu_7).$$

(6.18)

Solving these in a backward fashion we get $v_1 = P\{$the cut $\{3,5,9\}$ is the minimum capacity cut$\}$.

Forward Algorithm: Define

$$\bar{v}_j = P\{X_Y(t) = j \text{ for some } t \geq 0 \mid X_Y(0)=1\}, \quad 1 \leq j \leq m+1. \quad (6.19)$$

Then $R(Y) = \bar{v}_{m+1}$. The equations for $\bar{v}_j$ are given by

$$\bar{v}_j = \sum_{i<j} (q_{ij}(Y)/q_i(Y))\bar{v}_i, \quad (6.20)$$

which can be solved in a forward manner starting with $\bar{v}_1 = 1$ and computing $\bar{v}_2,\ldots,\bar{v}_{m+1} = R(Y)$ in that order.

7. Computational Results.

In this section we document the computational results using the networks shown in Figures 1,2,3 and 4. The networks are assumed to satisfy assumptions A1 and A2 of Section 4. The backward algorithms developed in Sections 5 and 6 are implemented in Fortran 77.

Table V gives the following network size descriptors for the four networks:

(1) Number of nodes,

(2) Number of arcs,

(3) Number of (s,t) paths,

(4) Number of (s,t) cuts, and

(5) Value of the uniformization constant q (defined by equation (5.9)).

Table VI gives the mean (μ) and standard deviation (σ) of the maximum flow in the four networks. The distribution of the maximum flow is computed using the uniformization technique described in Section 5. Thus, an integer M was chosen so that $\alpha_M \geq 1-10^{-5}$. The values of M and α_M are also given in Table VI. Note that one does not necessarily need larger M for larger networks.

The cumulative distribution function F(t) is approximated by $F_M(t)$ of equation (5.12). The distribution of the normalized maximum flow, i.e. (maximum flow-μ)/σ, is tabulated in Table VII. An interesting feature of this table is that the distribution of the normalized maximum flow seems to be more or less the same for all the four networks. No theoretical insight into this phenomenon is currently available, other than the intuitive

feeling that for "large" networks the distribution should be close to normal.

The conditional performance of the four networks is tabulated in the next four tables. For each network the tables list the following:

(1) R(Y), the probability that Y is the minimum capacity cut,

(2) $\mu(Y)$, the mean of the maximum flow given that Y is the minimum capacity cut,

(3) $\sigma(Y)$, the standard deviation of the maximum flow given that Y is the minimum capacity cut.

For network 1 all minimal cuts Y and conditional performance measures are tabulated in Table VIII. As the number of minimal cuts in network 2, 3 and 4 is large (17, 15 and 78 respectively) they are first ordered by decreasing criticality indices and only first ten are tabulated in Tables IX, X and XI respectively. Algorithm by Tsukiyama et al [1980] is used to enumerate all minimal (s,t) cuts in a network.

The algorithms proved to be very efficient and computationally stable. It took about one second of CPU time to compute the distribution of the maximum flow for each network and less than 0.2 seconds per cut to evaluate the conditional performance on IBM 4341 system. Using simulation to obtain equally accurate estimates would no doubt take considerably longer time.

8. Conclusion.

We have provided a method of computing the exact distribution of the maximum flow in planar networks with exponentially distributed arc capacities by constructing a continuous time Markov chain with one absorbing state so that the time until absorption in this state is equal to the maximum flow from source to sink in the network. The state space of this CTMC is shown to be the set of all paths in the network, augmented by an absorbing state. When the paths are put in "topmost path first" order the rate matrix of this chain becomes upper triangular. This makes all calculations very straightforward. Algorithms are developed to compute the distribution and moments of the maximum flow.

Further, a modified CTMC is constructed to compute the joint distribution of the criticality indicator of a given minimal (s,t) cut and the maximum flow. Algorithms are developed to compute the probability that the given cut is a minimum capacity cut in the network.

Note that it is possible to convert a maximum flow problem for a planar network with exponential arc capacities into a shortest path problem in the dual network with exponential arc lengths and solve this shortest path problem by the method described in Kulkarni [1984]. There are two disadvantages to this approach. First, one has to construct the dual, considerable extra work for large networks. Second, the state space requirement of this equivalent shortest path problem is almost always more than that of the path filling algorithm

described here. It is important to solve the maximum flow problem directly since it provides many interesting algorithms and concepts.

One major drawback of CTMC method presented here is that the size of the state space, which is equal to the number of paths, can grow exponentially with network size. Thus the time and space requirements of the algorithms can increase rapidly. It should be noted, however, that for very large networks, any other method of analysis will also be slow.

An extension of our method to non-planar networks does not seem possible, since the "topmost path first" order does not make sense in non-planar networks. It seems to us that an entirely different approach would be needed to solve the maximum flow problem in non-planar networks.

A quantity that is of interest in network flow analysis is the utilization of an arc, i.e. the fraction of the capacity utilized when the flow in the network is maximum. This and some related quantities will be studied in a forthcoming paper.

Table I. Paths in the Example Network in TPF Order

i	Nodes visited by path i
1	1 2 4 5
2	1 2 4 6 5
3	1 2 3 4 5
4	1 2 3 4 6 5
5	1 2 3 6 5
6	1 3 4 5
7	1 3 4 6 5
8	1 3 6 5

Table II. Q-matrix for the Example Network 1.

	1	2	3	4	5	6	7	8	9
1	$-\mu(P_1)$	μ_7	μ_3	0	0	μ_1	0	0	0
2	0	$-\mu(P_2)$	0	0	$\mu_3+\mu_8$	0	μ_1	0	μ_9
3	0	0	$-\mu(P_3)$	μ_7	μ_5	$\mu_1+\mu_4$	0	0	0
4	0	0	0	$-\mu(P_4)$	$\mu_5+\mu_8$	0	$\mu_1+\mu_4$	0	μ_9
5	0	0	0	0	$-\mu(P_5)$	0	0	$\mu_1+\mu_4$	$\mu_6+\mu_9$
6	0	0	0	0	0	$-\mu(P_6)$	μ_7	μ_5	μ_2
7	0	0	0	0	0	0	$-\mu(P_7)$	$\mu_5+\mu_8$	$\mu_2+\mu_9$
8	0	0	0	0	0	0	0	$-\mu(P_8)$	$\mu_2+\mu_6+\mu_9$
9	0	0	0	0	0	0	0	0	0

Notation: $\mu(P_i) = \sum\limits_{e \text{ on } P_i} \mu(e)$

Table III. The Y-admissible Paths
in Network 1.   Y = {3,5,9}.

i	Nodes visited by path i				
1	1	2	4	5	
2	1	2	3	4	5
3	1	2	3	6	5
4	1	3	4	5	
5	1	3	6	5	

Table IV.   The Q(Y) Matrix for Network 1.
Y = {3,5,9}

	1	2	3	4	5	6	7
1	$-\mu(P_1)$	μ_3	0	0	0	0	$\mu_1 + \mu_7$
2	0	$-\mu(P_2)$	μ_5	$\mu_1 + \mu_4$	0	0	μ_7
3	0	0	$-\mu(P_3)$	0	$\mu_1 + \mu_4$	μ_9	μ_6
4	0	0	0	$-\mu(P_4)$	μ_5	0	$\mu_2 + \mu_7$
5	0	0	0	0	$-\mu(P_5)$	μ_9	$\mu_2 + \mu_6$
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

Note: State 6 is the "good" absorbing state.
State 7 is the "bad" absorbing state.

Table V.   Network Size Descriptors

	Network 1	Network 2	Network 3	Network 4
# Nodes	6	7	7	10
# Arcs	9	11	11	25
# Paths	8	9	6	64
# Cuts	9	17	15	78
q	5.75	6.6666	4.5	5.0277

Table VI.   Mean and Std. Dev. of Maximum Flow

	Network 1	Network 2	Network 3	Network 4
M	24	32	19	80
α_M	0.999994	0.999993	0.999990	0.999991
μ	0.9964	1.1381	1.0236	3.9343
σ	0.5760	0.6279	0.6444	1.9013

Note: μ = mean of the maximum flow.
       σ = std. dev. of the maximum flow.

Table VII. $P((T-\mu)/\sigma \leq x)$

x	Network 1	Network 2	Network 3	Network 4
-3	0.0	0.0	0.0	0.0
-2	0.0	0.0	0.0	0.0
-1	0.1391	0.1421	0.1354	0.1487
0	0.5727	0.5680	0.5788	0.5568
1	0.8502	0.8496	0.8506	0.8473
2	0.9566	0.9574	0.9554	0.9596
3	0.9887	0.9893	0.9880	0.9910

Table VIII.   Conditional Performance: Network 1.

Cut #	$Y = C(X,\overline{X})$	$R(Y)$	$\mu(Y)$	$\sigma(Y)$
1	3,5,6	0.2678	1.0409	0.5787
2	1,2	0.1780	0.8949	0.5632
3	2,3,4	0.1608	1.0461	0.5765
4	6,7,8	0.1382	1.0554	0.5785
5	7,9	0.1142	0.9056	0.5657
6	1,5,6	0.0680	0.9989	0.5683
7	3,5,9	0.0535	1.0056	0.5698
8	1,5,9	0.0136	0.9636	0.5588
9	2,4,7,8	0.0059	1.0138	0.5154

Table IX. Conditional Performance: Network 2.

Cut #	Y = C(X,$\overline{X}$)	R(Y)	μ(Y)	σ(Y)
1	1,2	0.2204	0.9776	0.6049
2	8,9,11	0.1405	1.0833	0.6060
3	1,6,7	0.0923	1.1483	0.6263
4	3,9,11	0.0878	1.0833	0.6060
5	7,8,9,10	0.0798	1.3066	0.6409
6	4,6,7,8	0.0770	1.3066	0.6409
7	2,4,5,8	0.0722	1.2394	0.6164
8	3,7,9,10	0.0499	1.3066	0.6409
9	3,4,6,7	0.0481	1.2990	0.6393
10	2,3,4,5	0.0451	1.2394	0.6164

Table X. Conditional Performance: Network 3.

Cut #	Y = C(X,$\overline{X}$)	R(Y)	μ(Y)	σ(Y)
1	1,2	0.3051	0.9899	0.6516
2	1,6	0.1566	0.8467	0.5590
3	7,8,11	0.1045	1.0708	0.6292
4	3,8,11	0.0871	1.0708	0.6292
5	1,11	0.0687	0.7289	0.4947
6	4,6,7	0.0621	1.1128	0.6337
7	6,7,8,9	0.0529	1.3103	0.6712
8	3,6,8,9	0.0441	1.3103	0.6712
9	4,6,7	0.0211	0.9790	0.5807
10	3,4,11	0.0175	0.9790	0.5807

Table XI. Conditional Performance: Network 4.

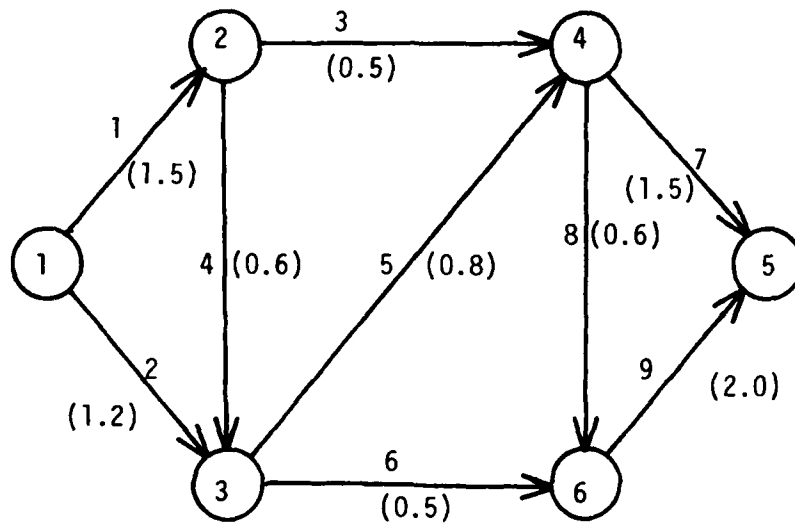| Cut #	Y = C(X,$\overline{X}$)	R(Y)	μ(Y)	σ(Y)
1	1,2,3	0.2819	3.5613	1.8450
2	13,19,25	0.2451	3.6926	1.8683
3	1,2,11,12	0.0649	3.9192	1.8378
4	1,3,7,8,9	0.0470	4.4759	1.9083
5	12,13,19,22	0.0374	4.0378	1.8546
6	3,9,13,19,20	0.0373	4.2483	1.8515
7	3,4,5,8,9	0.0330	4.4151	1.8845
8	1,7,8,9,12	0.0289	4.4550	1.8916
9	2,3,4,5	0.0228	3.6346	1.7365
10	4,15,19,25	0.0226	4.1453	1.8847

Figure 1. The Example Network 1.

The numbers on the arcs are arc numbers.
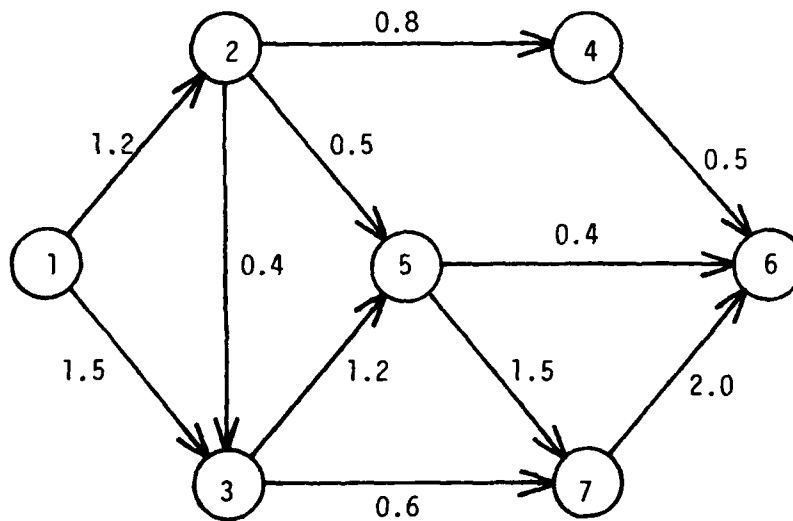The numbers in brackets represent mean capacities.

Figure 2.   The Example Network 2.

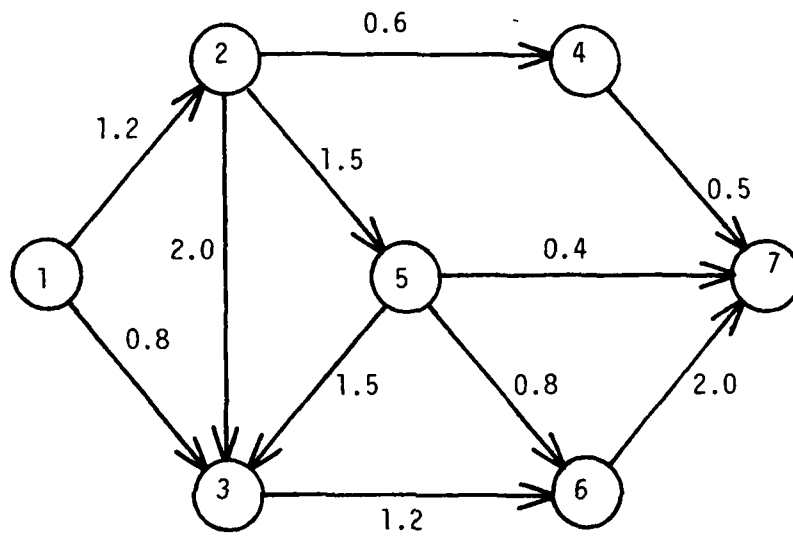The numbers on the arcs represent mean capacities.

Figure 3.   The Example Network 3.

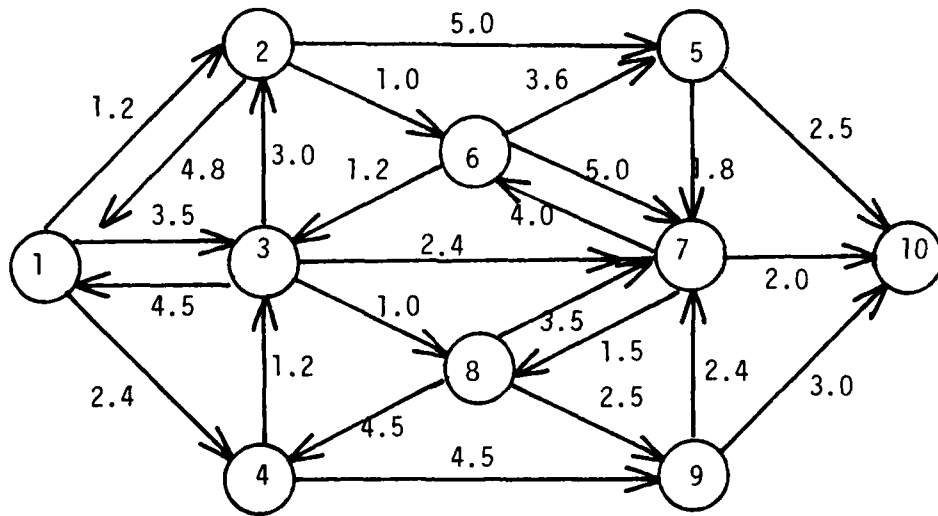The numbers on the arcs represent mean capacities.

Figure 4.   The Example Network 4.

The numbers on the arcs represent mean capacities.

References:

1. Berge, C. and Ghouilla-Houri, A. (1962), Programming, Games and Transportation Networks, Wiley, New York.

2. Dijkstra, E.W. (1959), "A Note on Two Problems in Connexion with Graphs", Numberische Mathematik, 1, pp. 269-271.

3. Elmaghraby, S.E. (1977), Activity Networks: Project Planning and Control by Network Models, John Wiley and Sons.

4. Evans, J.R. (1976), "Maximum Flow in Probabilistic Graphs - Discrete Case", Networks, 6, pp. 161-183.

5. Ford, L.R. Jr. and Fulkerson, D.R. (1956), "Maximal Flow Through A Network", Canadian Journal of Mathematics, 8, pp. 399-404.

6. Ford, L.R. Jr. and Fulkerson, D.R. (1962), Flows in Networks, Princeton University Press, Princeton.

7. Frank, H. and Frisch, I.T. (1971), Communication, Transmission, and Transportation Networks, Addison-Wesley.

8. Itai, A. and Shiloach, Y. (1979), "Maximum Flow in Planar Networks", SIAM J. Comput. 8, No. 2, pp. 135-150.

9. Kulkarni, V.G. and Adlakha, V.G. (1984), "Markov and Markov Regenerative PERT Networks", TR NO. 84-4, Curriculum in ORSA, UNC, Chapel Hill, N.C. 27514.

10. Kulkarni, V.G. (1984), "Shortest Paths in Markov Networks", TR No. 84-10, Curriculum in ORSA, UNC, Chapel Hill, N.C. 27514.

11. Ross, S.M. (1983), Stochastic Processes, Academic Press.

12. Reed, R.C. and Tarjan, R.E. (1975), "Bounds on Backtrack Algorithms for Listing Cycles, Paths and Spanning Trees", Networks, 5, pp. 237-252.

13. Tsukiyama, S., Shirakawa, I., Ozaki, H. and Ariyoshi, H. (1980), "An Algorithm to Enumerate all Cutsets of a Graph in Linear Time per Cutset", JACM, 27, pp. 619-632.

END

FILMED

5-85

DTIC